WHAT IS CLAIMED IS:

1. A method of processing information represented by an original series of (run, level) pairs, said method comprising:

a) inspecting the (run, level) pairs in the original series of (run, level) pairs to determine whether or not modification of at least one (run, level) pair in the original series of (run, level) pairs would produce a desirable decrease in a number of bits required for variable-length encoding of said information despite introduction of noise into the variable-length encoding of said information; and

b) upon determining that modification of said at least one (run, level) pair in the original series of (run, level) pairs would produce a desirable decrease in the number of bits required for variable-length encoding of said information despite introduction of noise into the variable-length encoding of said information, modifying said at least one (run, level) pair to produce a modified series of (run, level) pairs from the original series of (run, level) pairs; and

c) variable-length encoding the modified series of (run, level) pairs.

2. The method as claimed in claim 1, which is performed by sequentially inspecting each (run, level) pair to determine whether or not modification of said each (run, level) pair would produce a desirable decrease in the number of bits required for variable-length

encoding of said information despite introduction of noise into the variable-length encoding of said information; and if modification of said each (run, level) pair would produce a desirable decrease in the number of bits required for variable-length encoding of said information despite introduction of noise into the variable-length encoding of said information, then modifying said each (run, level) pair; and then variable-length encoding said each (run, level) pair.

3. The method as claimed in claim 1, wherein the inspecting the (run, level) pairs in the original series of (run, level) pairs includes lookup of a table specifying whether or not certain (run, level) pairs should be modified.

4. The method as claimed in claim 1, wherein the inspecting the (run, level) pairs in the original series of (run, level) pairs includes testing for certain ranges of run lengths and level values to determine whether or not certain (run, level) pairs should be modified.

5. The method as claimed in claim 1, wherein said at least one (run, level) pair has a run length of M that is greater than zero and a level value of N, and the production of the modified series of (run, level) pairs from the original series of (run, level) pairs includes substituting, for said at least one (run, level) pair, a first (run, level) pair immediately followed by a second (run, level) pair, the first (run, level) pair having a run length of M-1 and a level having a minimum non-zero magnitude, the second (run, level) pair having a run length of zero and a level value of N.

6. The method as claimed in claim 5, which includes decoding the variable-length encoding of the modified series of (run, level) pairs to produce a decoded series of (run, level) pairs, and inspecting the (run, level) pairs in the decoded series of (run, level) pairs to find the first (run, level) pair having a minimum non-zero magnitude immediately followed by the second (run, level) pair having a run length of zero, and determining that the first (run, level) pair is likely to be noise introduced during the production of the modified series of (run, level) pairs from the original series of (run, level) pairs and therefore rejecting the first (run, level) pair.

7. The method as claimed in claim 6, which includes a table lookup for determining that the first (run, level) pair is likely to be noise introduced during the production of the modified series of (run, level) pairs from the original series of (run, level) pairs.

8. The method as claimed in claim 1, which includes decoding the variable-length encoding of the modified series of (run, level) pairs to produce a decoded series of (run, level) pairs, and inspecting the (run, level) pairs in the decoded series of (run, level) pairs to find (run, level) pairs likely to be noise introduced during the production of the modified series of (run, level) pairs from the original series of (run, level) pairs and therefore rejecting the (run, level) pairs likely to be noise introduced during the production of the modified series of (run, level) pairs from the original series of (run, level) pairs.

9. A method of variable-length encoding a block of pixels, the method comprising:

a) computing a two-dimensional discrete cosine transform (DCT) of the block of pixels to produce a series of DCT coefficient values;

b) quantizing the DCT coefficient values to produce quantized coefficient values;

c) producing an original series of (run, level) pairs each having a level value indicating a respective non-zero quantized coefficient value;

d) inspecting the (run, level) pairs in the original series of (run, level) pairs to determine whether or not modification of at least one (run, level) pair in the original series of (run, level) pairs would produce a desirable decrease in a number of bits required for variable-length encoding of said block of pixels despite introduction of noise into the variable-length encoding of said block of pixels; and

e) upon determining that modification of said at least one (run, level) pair in the original series of (run, level) pairs would produce a desirable decrease in the number of bits required for variable-length encoding of said block of pixels despite introduction of noise into the variable-length encoding of said block of pixels, modifying said at least one (run, level) pair to produce a modified series of (run, level) pairs from the original series of (run, level) pairs; and

f) variable-length encoding the modified series of (run, level) pairs.

10. The method as claimed in claim 9, which is performed by sequentially inspecting each (run, level) pair to determine whether or not modification of said each (run, level) pair would produce a desirable decrease in the number of bits required for variable-length encoding of said block of pixels despite introduction of noise into the variable-length encoding of said block of pixels; and if modification of said each (run, level) pair would produce a desirable decrease in the number of bits required for variable-length encoding of said block of pixels despite introduction of noise into the variable-length encoding of said block of pixels, then modifying said each (run, level) pair; and then variable-length encoding said each (run, level) pair.

11. The method as claimed in claim 9, wherein the inspecting of the (run, level) pairs in the original series of (run, level) pairs includes lookup of a table specifying whether or not certain (run, level) pairs should be modified.

12. The method as claimed in claim 9, wherein the inspecting of the (run, level) pairs in the original series of (run, level) pairs includes testing for certain ranges of run lengths and level values to determine whether or not certain (run, level) pairs should be modified.

13. The method as claimed in claim 9, wherein said at least one (run, level) pair has a run length of M that is greater than zero and a level value of N, and the production of the modified series of (run, level) pairs from the original series of (run, level) pairs includes

substituting, for said at least one (run, level) pair, a first (run, level) pair immediately followed by a second (run, level) pair, the first (run, level) pair having a run length of M-1 and a level having a minimum non-zero magnitude, and the second (run, level) pair having a run length of zero and a level value of N.

14. The method as claimed in claim 9, wherein the production of the original series of (run, level) pairs from the quantized DCT coefficient values includes identifying some DCT coefficients having non-zero values that are less significant than values of other DCT coefficients, the original series of (run, level) pairs does not include (run, level) pairs encoding level values for said some DCT coefficients, said first (run, level) pair specifies a level value for one of said some DCT coefficients, said one of said some DCT coefficients has a sign, and the level value of said first (run, level) pair is selected to have the same sign as the sign of said one of said some DCT coefficients.

15. The method as claimed in claim 9, wherein the production of the original series of (run, level) pairs from the quantized DCT coefficient values includes identifying some DCT coefficients having non-zero values that are less significant than values of other DCT coefficients, the original series of (run, level) pairs does not include (run, level) pairs encoding level values for said some DCT coefficients, and the method includes modifying at least one (run, level) pair in order to reduce noise without increasing the number of bits for the variable-length encoding by including in the modified series a (run, level) pair encoding a minimum magnitude level for at least one of said some DCT coefficients, said at least one of said some DCT coefficients has a sign, and the (run,

level) pair encoding a minimum magnitude level for said at least one of said some DCT coefficients has a sign equal to the sign of the said at least one of said some DCT coefficients.

16. A method of producing MPEG encoded video from an original series of MPEG-compliant (run, level) pairs, said method comprising:

a) inspecting the (run, level) pairs in the original series of (run, level) pairs to determine whether or not modification of at least one (run, level) pair in the original series of (run, level) pairs would produce a desirable decrease in a number of bits in the MPEG encoded video despite introduction of noise into the MPEG encoded video; and

b) upon determining that modification of said at least one (run, level) pair in the original series of (run, level) pairs would produce a desirable decrease in the number of bits in the MPEG encoded video despite introduction of noise into the MPEG encoded video, replacing said at least one (run, level) pair with a sequence of a first (run, level) pair and a second (run, level) pair to produce a modified series of (run, level) pairs from the original series of (run, level) pairs, said at least one (run, level) pair having a non-zero run length of M and a non-zero level value of N, the first (run, level) pair having a run length of M-1 and a level magnitude of one, and the second (run, level) pair having a run length of zero and a level value of N; and

c) variable-length encoding the modified series of (run, level) pairs to produce the MPEG encoded video.

17. The method as claimed in claim 16, which includes sequentially inspecting each (run, level) pair in the original series of MPEG-compliant (run, level) pairs to determine whether or not modification of said each (run, level) pair would produce a desirable decrease in the number of bits in the MPEG encoded video despite introduction of noise into the MPEG encoded video; and if modification of said each (run, level) pair would produce a desirable decrease in the number of bits required in the MPEG encoded video despite introduction of noise into the MPEG encoded video, then modifying said each (run, level) pair; and then variable-length encoding said each (run, level) pair.

18. The method as claimed in claim 16, wherein the inspecting of the (run, level) pairs in the original series of MPEG-compliant (run, level) pairs includes lookup of a table specifying whether or not certain (run, level) pairs should be modified.

19. The method as claimed in claim 16, wherein the inspecting of the (run, level) pairs in the original series of (run, level) pairs includes testing for certain ranges of run lengths and level values to determine whether or not certain (run, level) pairs should be modified.

20. A method of decoding MPEG encoded video that includes noise introduced during the encoding process by insertion of at least one (run, level) pair having a level magnitude of one, said method comprising;

a) decoding a series of (run, level) pairs from the MPEG encoded video; and

b) inspecting the series of (run, level) pairs to find said at least one (run, level) pair having a level magnitude of one; and

c) determining that said at least one (run, level) pair having a level magnitude of one is likely to represent noise introduced during the encoding process, and therefore rejecting said at least one (run, level) pair having a level magnitude of one in order to reduce noise.

21. The method as claimed in claim 20, which includes a table lookup using the run length of said at least one (run, level) pair having a level magnitude of one and a level magnitude of a (run, level) pair immediately following said at least one (run, level) pair having a level magnitude of one for determining that said at least one (run, level) pair having a level magnitude of one is likely to represent noise introduced during the encoding process.

22. The method as claimed in claim 20, wherein the noise is introduced during the encoding process by insertion of (run, level) pairs having a predetermined level value having a magnitude of one, and the decoding process does not reject (run, level) pairs having a level value different from the predetermined level value.

23. A digital computer for producing MPEG encoded video from an original series of MPEG-compliant (run, level) pairs, said digital computer comprising at least one processor programmed for:

a) inspecting the (run, level) pairs in the original series of (run, level) pairs to determine whether or not modification of at least one (run, level) pair in the original series of (run, level) pairs would produce a desirable decrease in a number of bits in the MPEG encoded video despite introduction of noise into the MPEG encoded video; and

b) upon determining that modification of said at least one (run, level) pair in the original series of (run, level) pairs would produce a desirable decrease in the number of bits in the MPEG encoded video despite introduction of noise into the MPEG encoded video, replacing said at least one (run, level) pair with a sequence of a first (run, level) pair and a second (run, level) pair to produce a modified series of (run, level) pairs from the original series of (run, level) pairs, said at least one (run, level) pair having a non-zero run length of M and a non-zero level value of N, the first (run, level) pair having a run length of M-1 and a level magnitude of one, and the second (run, level) pair having a run length of zero and a level value of N; and

c) variable-length encoding the modified series of (run, level) pairs to produce the MPEG encoded video.

24. The digital computer as claimed in claim 23, wherein said at least one processor is programmed for sequentially inspecting each (run, level) pair in the original series of MPEG-compliant (run, level) pairs to determine whether or not modification of said each (run, level) pair would produce a desirable decrease in the number of bits required in the MPEG encoded video despite introduction of noise into the MPEG encoded video; and if modification of said each (run, level) pair would produce a desirable decrease in the number of bits required for variable-length encoding of the MPEG encoded video despite introduction of noise into the MPEG encoded video, then modifying said each (run, level) pair; and then variable-length encoding said each (run, level) pair.

25. The digital computer as claimed in claim 23, wherein said at least one processor is programmed for lookup of a table specifying whether or not certain (run, level) pairs should be modified.

26. The digital computer as claimed in claim 23, wherein said at least one processor is programmed for testing for certain ranges of run lengths and level values to determine whether or not certain (run, level) pairs should be modified.

27. A decoder for decoding MPEG encoded video that includes noise introduced during the encoding process by insertion of at least one (run, level) pair having a level magnitude of one, the decoder comprising at least one processor programmed for:

a) decoding a series of (run, level) pairs from the MPEG encoded video; and

b) inspecting the (run, level) pairs to find said at least one (run, level) pair having a level magnitude of one; and

c) determining that said at least one (run, level) pair having a level magnitude of one is likely to represent noise introduced during the encoding process, and therefore rejecting said at least one (run, level) pair having a level magnitude of one in order to reduce noise.

28. The decoder as claimed in claim 27, wherein said at least one processor is programmed to perform a table lookup using the run length of said at least one (run, level) pair having a level magnitude of one and a level magnitude of a (run, level) pair immediately following said at least one (run, level) pair having a level magnitude of one for determining that said at least one (run, level) pair having a level magnitude of one is likely to represent noise introduced during the encoding process.

29. The decoder as claimed in claim 27, wherein the noise is introduced during the encoding process by insertion of (run, level) pairs having a predetermined level value having a magnitude of one, and said at least one processor is programmed not to reject (run, level) pairs having a level value different from the predetermined level value.